Cite No. 1

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 December 2002 (27.12.2002)

PCT

(10) International Publication Number
WO 02/103999 A2

(51) International Patent Classification[7]: H04N

(21) International Application Number: PCT/US02/19160

(22) International Filing Date: 16 June 2002 (16.06.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
101 29 108.6 16 June 2001 (16.06.2001) DE

(71) Applicant *(for all designated States except US)*: **HARMAN INTERNATIONAL INDUSTRIES, INC.** [US/US]; 8500 Balboa Boulevard, Northridge, CA 91329 (US).

(71) Applicants and
(72) Inventors: **NITZPON, Hans-Juergen** [DE/DE]; Steinaeckerstrasse 3, D-76337 Waldbronn (DE). **KLAUS-WAGENBRENNER, Jochen** [DE/DE]; Am Bergwald 3, D-76571 Gaggenau (DE). **TEICHNER, Detlef** [DE/DE]; Ortinstrasse 51, D-18126 Koenigsfeld (DE).

(74) Agent: **RUBIO-CAMPOS, Francisco, A.**; Sonnenschein Nath & Rosenthal, P.O. Box #061080, Wacker Drive Station, Sears Tower, Chicago, IL 60606-1080 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 02/103999 A2

(54) Title: BIT STREAM CONVERSION SYSTEM

(57) Abstract: A bit-stream converter capable of converting a first synchronous compressed bit-stream of data at a first sampling rate to second synchronous compressed bit-stream frame of data at a second sampling rate is disclosed. The bit-stream converter architecture may include a payload length detector and a zero stuffing unit in signal communication with the payload length detector. The zero stuffing unit is capable of zero stuffing section responsive to the payload length detector detecting the payload length.

BIT STREAM CONVERSION SYSTEM

1. Field of the Invention.

This invention relates generally to the field of data communications. In particular, the invention relates to data communication systems that utilize compressed bit-streams.

2. Related Art.

The complexity of consumer video and audio electronics component systems is increasing at rapid pace. Systems such as compact disk ("CD"), laserdisc, digital video disc or digital versatile disc ("DVD"), mini-disk, and others are now common. As a result, a modern trend is to integrate all these systems into home theater and automotive entertainment systems.

Generally, current CD and DVD chipsets provide a Sony/Philips Digital Interface ('S/PDIF") that outputs audio in bit-streams of data according to the ISO/IEC 60958 (i.e., linear pulse code modulation ("PCM")) and ISO/IEC 61937 (i.e., non-linear PCM) standards. Typically, compressed multi-channel audio bit-streams, such as Dolby Digital® (AC-3), DTS®, MLP®, MP3®, MPEG II®, MPEG II-AAC® etc. are formatted according to ISO/IEC 61937 and are conveyed over S/PDIF to an external audio decoder. The bit-streams of data are bi-phased coded with a symbol frequency of 64 times (for very low bit-rates 128 times) the original sampling-rate ("f_{sample}"). A S/PDIF receiver typically locks on the bit-stream and synchronously generates the f_{sample} for decoding the bit-stream of data. The sampling

frequency of the original bit-stream may cover the range of 8 – 192 kHz (e.g., CD are typically 44.1 kHz, DVD-V typically 48 kHz, DVD-A typically 96 kHz, etc.).

Multimedia networks may be utilized to integrate CD and DVD type components into modern home theater and automotive entertainment systems. Unfortunately, many multimedia networks operate in a synchronous manner at a constant rate of e.g. 44.1 kHz that is different than the encoded audio source. In order to transport digital audio from a digital source (such as a CD or DVD) to a destination (such as a decoding amplifier) over the synchronous channels, the audio sampling rate (e.g. 48 kHz for a DVD) needs to be adapted to the multimedia network sampling rate (44.1 kHz). A previous approach to adapt the two sampling rates includes sample rate converting the audio. However, since compressed multi-channel audio is a bit-stream rather than pulse code modulation ("PCM") samples, this approach cannot be applied immediately. The audio needs to be decoded first into typically 5.1 PCM channels and then sample rate conversion may be applied prior to sending it over multimedia network. Decoded audio, however, occupies much more bandwidth than the compressed bit-stream. Therefore, there is a need for a system that is capable of adapting the two sampling rates.

SUMMARY

This invention provides a bit-stream converter capable of converting a first synchronous compressed bit-stream frame of data at a first sampling rate to a second synchronous compressed bit-stream frame of data at a second sampling rate. Such a bit-stream converter may utilize a system architecture that performs a process for converting a first synchronous compressed bit-stream frame of data at a first sampling rate to a second synchronous compressed bit-stream frame of data at a second sampling rate. The process may include determining a format for the first compressed bit-stream frame. The first compressed bit-stream frame may have a frame length and may include a data-burst section and a stuffing section. The data-burst section may have a payload section including a preamble section and a payload length, while the stuffing section may have a stuffing length. The process may also include zero stuffing the stuffing section in response to a particular format.

The bit-stream converter architecture may include a payload length detector and a zero stuffing unit in signal communication with the payload length detector. The zero stuffing unit is capable of zero stuffing the stuffing section responsive to the payload length detector detecting the payload length.

This invention also provides an inverse bit-stream converter for converting a first synchronous compressed bit-stream frame of data at a first sampling rate having zero stuffing to a second synchronous compressed bit-stream frame of data at a second sampling rate. The bit-stream converter may include a synchronization unit, a format detector in signal communication with the synchronization unit and a zero stuffing removal unit in signal communication with format detector. The format

detector may be capable of determining a format for the first synchronous compressed bit-stream frame of data.

Other systems, methods, features and advantages of the invention will be or will become apparent to one with skill in the art upon examination of the following figures and detailed description. It is intended that all such additional systems, methods, features and advantages be included within this description, be within the scope of the invention, and be protected by the accompanying claims.

BRIEF DESCRIPTION OF THE FIGURES

The components in the figures are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention. In the figures, like reference numerals designate corresponding parts throughout the different views.

FIG. 1 is a block diagram illustrating an example implementation of bit-stream conversion system.

FIG. 2 is a block diagram illustrating an example format of a bit-stream.

FIG. 3 is a block diagram illustrating an example implementation of the bit-stream converter element of FIG. 1.

FIG. 4 is a block diagram illustrating an example implementation of the inverse bit-stream converter element of FIG. 1.

FIG. 5 is a flowchart illustrating an example process preformed by the bit-stream converter of FIG. 3.

FIG. 6 is a flowchart illustrating an example process preformed by the inverse bit-stream converter of FIG. 4.

DETAILED DESCRIPTION

FIG. 1 shows a multimedia data communication system 100 including the bit-stream conversion system 102. The bit-stream conversion system 102 may include a bit-stream converter 104 in signal communication with a source 106 and a network 108. The inverse bit-stream converter 110 may be in signal communication with the network 108 (such as a multimedia network) and a decoder 112.

The source 106 may be a compact disk ("CD") or derivative product, a mini-disc or derivative product, a digital video disc or digital versatile disc ("DVD") or derivative product, or other equivalent type sources. The network 108 may be any link or network (wireless or physical link) that provides a clock (i.e. being clock master) that is different from the S/PDIF source.

The bit-stream conversion system 102 converts compressed bit-streams of data from the source 106 to match the network 108 sampling rate (also known as the transport rate) without altering the audio information in the bit-stream converter 104. The inverse bit-stream converter 110 then receives converted compressed bit-streams of data from the network 108 and determines the original sampling frequency f_{sample} 114 and outputs new bit-stream of data 116 that is a reproduction of the bit-stream of data produced by the source 106. The new bit-stream of data 116 is input into the decoder 112 and the decoder 112 decodes the new bit-stream of data 116 producing separate pulse coded modulation ("PCM") channels that may be transmitted to a receiver via signal path 120.

FIG. 2 is a block diagram illustrating an example format of a bit-stream 200. The bit-stream 200 may include numerous frames 202. Each frame 202 may include

sub-frames such as a data-burst section 204 and stuffing section 206. The data-burst section may include a preamble and payload section 208. The preamble may include header information such as Pa 210, Pb 212, Pc 214 and Pd 216. Pa may equal 0xF872 and Pb may equal 0x4E1F. Both Pa and Pb represent a synchronization word that indicates the start of the data burst and may be utilized to obtain the sampling rate f_{sample}. Pc represents the burst information and indicates the type of data in the bit-stream and some information and/or control for the receiver (not shown). Pd represents the length of the burst-payload in bits. The frame 202 has a period T_{period} 218.

As an example, if the network 108 is designed to transmit CD audio signals, the network 108 may operate with a sampling frequency f_{sample} of approximately 44.1 kHz and may be designed to transmit two channel linear PCM signals at 44.1 kHz. If the source 106 is a DVD, instead of a CD, the source 106 may transmit bit-streams of data that include multi-channel audio signals. These multi-channel audio signals may be compressed such that their transmission rate is lower than their equivalent 2-channel PCM version. In this methodology, multi-channel audio can be transmitted utilizing less than or equal to the same channel bandwidth of linear stereo PCM. As a result, the data length of the payload section of the DVD signal will be shorter than the equivalent data length of the payload section of a CD signal. Therefore, in order to maintain the same transmission signal period T_{period} 218 between the DVD and CD signals, zero stuffing may be utilized to expand the length of the stuffing section 206 in order to compensate for the shorter payload section 208.

For example, IEC 61937 specifies how non-linear PCM (compressed audio) is transferred over S/PDIF. S/PDIF is a unidirectional bi-phased coded link and there is no handshake between source 106 and the destination. The compressed audio frame always represents a constant number of samples, (1536 for Dolby Digital® AC-3). According to the compression rate, the actual data burst may be shorter (i.e., a high compression rate) or longer (i.e., a low compression rate). However, since there is no handshaking in S/PDIF, the process clocks out the data frame, which in this example is 1536 x (64 x sampling frequency) clock periods, before the next data burst is sent. Since the payload is lower than 1536 x (64 x f_{sample}), the rest is filled with zero bits ("zero stuffing").

By reducing and/or stretching the zero-stuffing, the burst-payloads may be transported at a different data rate without affecting the payload. In a typical Dolby Digital® bit-stream the sampling frequency is 48 kHz and the compression rate is 448 kbps. One compressed Dolby Digital® frame always represents 1536 samples. The original repetition rate between 2 data bursts is, therefore, 1536/48 kHz = 32 ms. If the network is operating at 44.1 kHz, the repetition rate equivalently needs to be reduced to 1411.2 in order not to loose any information (1411.2/44.1 kHz = 32 ms). Consequently, the amount of zero-stuffing should be reduced by 124.8 IEC 60958/61937 frames. Because 1411.2 is a rational number, the goal is to reduce the stuffing of 4 consecutive burst-payloads by 125 frames (1411) and the 5th burst-payload by 124 IEC 60958/61937 frames (1412), such that the average data rate of 1411.2 is respected.

In this example, the original frame repetition rate is 32 ms (burst-payload and stuffing). However, for a network clock (e.g. 44.1 kHz) that is lower than the source clock (e.g. 48 kHz), less bits need to be transported before starting the next frame. Therefore, the amount of zeros should be reduced because it does not affect the payload. The amount of reduction is represented by the relation of 48 / 44.1. Because this relation is not an integer, an approach is applied that is similar to a leap-year correction. Here, every fifth frames is slightly longer so that the average frame rate remains 32 ms. If the source is at a lower f_{sample} (say 32 kHz) than the network, then the amount of zeros has to be increased (stretched) correspondingly.

For other formats, the compression may be relatively low. For example, the DTS format has 6 IEC 60958/61937 zero-frames available between 2 burst-payloads. This is less than required for bit-stream conversion from 48 kHz to 44.1 kHz. Therefore, in this example, a 2nd stereo transport channel may be utilized to transport all information at 44.1 kHz (assume DTS 48 kHz, bit-rate = 1509.75 kbps). The following table summarizes some typical examples:

Format	Preamble Pc	IEC 61937 Repetition Period Frames (bytes)	Bit-rate (kbps)/ Payload (bytes)	Network Repetiton Period for 44.1 kHz
AC-3 (48kHz)	1	1536 (6144)	(32-640) / (128-2560)	1411.2 (4x1411 + 1x1412)
MP3 (32kHz)	5	1152 (4608)		1587.6 (4x1588 + 1x1586)
MP3 (44.1kHz)	5	1152 (4608)		1152
MP3 (48kHz)	5	1152 (4608)		1058.4 (4x1058 + 1x1060)
AAC (48kHz)	7	1024 (4096)		940.8 (4x941 + 1x940)
DTS I (48kHz)	11	512 (2048)	754.50 / 1006	470.4 (4x470 + 1x472)
DTS II (44.1kHz)	12	1024 (4096)	1234.00 / 4096	1024
DTS I (48kHz)	11	512 (2048)	1509.75 / 2013	470.4 (4x470 + 1x472)
DTS III (24/96)	13	2048 (8192)		940.8 (4x941 + 1x940)

FIG. 3 is a block diagram illustrating an example implementation of the bit-stream converter 104. The bit-stream converter 104 may include a synchronization unit 300, a frequency detector 302, a payload length detector 304, a zero stuffing unit 306 and a counter 308. The synchronization unit 300 is in signal communication with the source 106 via a signal path 310. The synchronization unit 300 is also in signal communication with frequency detector 302 and counter 308. Frequency detector 302 is in signal communication with both the synchronization unit 300 and the payload length detector 304. The zero stuffing unit 306 is in signal communication with the payload length detector 304, the counter 308 and the network 108 via signal path 312. The counter 308 may be a modulo-N counter (in this case N=5).

In operation, the synchronization unit 300 (also known as "SYNC") identifies the preamble Pa, Pb, Pc and Pd of a new burst-payload. The SYNC compares the bit-stream to the preamble Pa and Pb, and if a match is found, the SYNC triggers the modulo-N (here N = 5) counter for correct zero stuffing modification at the zero stuffing unit 306. Pc may act to identify the type of encoding. The SYNC reads the Pc, the frequency detector 302 detects the sampling frequency and the length of the payload is determined by the payload length detector (from reading Pd) in order to modify the zero stuffing by the zero stuffing unit 306. This methodology also may determine how many network channels need to be allocated in parallel.

FIG. 4 is a block diagram illustrating an example implementation of the inverse bit-stream converter 110. The inverse bit-stream converter 110 may include a synchronization unit 400, a format detector 402, a frequency detector 404 and a phase lock loop ("PLL") 406. The synchronization unit 400 is in signal communication with the network 108, the format detector 402 and the frequency detector 404. The frequency detector 404 is in signal communication with the PLL 406 and the decoder 112.

In operation, the inverse bit-stream converter 110 extracts the original bit-stream information and triggers the PLL 406 to generate synchronously the original sampling frequency 114. For example, if the network clock 420 is operating at 44.1

kHz but the frequency detector 404 detects that the original bit-stream is at 48 kHz, the PLL 406 is driven by the network clock 420 and the frequency detector 404 to recover the 48 kHz required by the decoder 112. The decoder 112 uses the parameters from 402 and 404 to properly decode the audio in order to produce and output a signal via signal path 120.

A controller (not shown) may be utilized to control the operation of the bit-stream converter 104 and inverse bit-stream converter 110. The controller may be any type of control device that may be selectively implemented in software, hardware (such as a computer, processor, micro controller or the equivalent), or a combination of hardware and software. The controller may utilize optional software (not shown).

The software, includes an ordered listing of executable instructions for implementing logical functions, may selectively be embodied in any computer-readable (or signal-bearing) medium for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that may selectively fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. In the context of this document, a "computer-readable medium" and/or "signal-bearing medium" is any means that may contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device. The computer readable medium may selectively be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific examples "a non-exhaustive list" of the computer-readable medium would include the

following: an electrical connection "electronic" having one or more wires, a portable computer diskette (magnetic), a RAM (electronic), a read-only memory "ROM" (electronic), an erasable programmable read-only memory (EPROM or Flash memory) (electronic), a Magnetic Random Access Memory ("MRAM"), a Ferro Random Access Memory ("FRAM"), a chalcogenide memory or Ovonic Universal Memory ("OUM"), a polymer memory, a MicroElectroMechanical (MEMS") memory and a write once 3D memory, an optical fiber (optical), and a portable compact disc read-only memory "CDROM" (optical). Note that the computer-readable medium may even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via for instance optical scanning of the paper or other medium, then compiled, interpreted or otherwise processed in a suitable manner if necessary, and then stored in a computer memory.

FIG. 5 is a flowchart 500 illustrating an example process performed by the bit-stream converter 104. This process may be performed by hardware, software or combination of both. The process starts 502 with the input reception 504 of information such as a bit-stream of data by bit-stream converter 104. The synchronization unit 300 determines the preamble values Pa, Pb, Pc and Pd 506. In decision 508, a comparator unit (not shown) within the synchronization unit 300 compares the bit-stream to the preamble parameters Pa and Pb 508. If the result in the decision 508 is not an approximate match between the bit-stream and preamble values Pa and Pb, the process returns 509 to step 504 and repeats.

If instead the result of decision 508 is an approximate match between the bit-stream and preamble values Pa and Pb, the counter is started 510 and the sampling frequency of the bit-stream is determined 512. Next, the payload length detector 304 determines the payload length 514. Next, the zero stuffing unit stuffs the stuffing section with the appropriate number of zeros 516 and the process ends at step 518.

FIG. 6 is a flowchart 600 illustrating an example process performed by the inverse bit-stream converter 110. The example process may be performed by hardware, software or combination of both. The process starts at step 602 with the input and reception of the bit-stream data 604 by the inverse bit-stream converter 110. The synchronization unit 400 determines the preamble values Pa, Pb, Pc and Pd 606. In decision 608, a comparator unit (not shown) within the synchronization unit 400 compares the bit-stream to the preamble parameters Pa and Pb. If the result in decision step 608 is not an approximate match between the bit-stream and preamble values Pa and Pb, the process returns 610 to step 604 and repeats.

If instead the result of decision 608 is an approximate match between the bit-stream and preamble values Pa and Pb, the format detector determines the format type Pc of the bit-stream 612. The frequency detector then determines the original sampling frequency of the compressed audio 614. Next, the decoder 112 decodes the bit-stream 616 and produces an output signal that is transmitted to a receiver, e.g. a digital to analog converter (not shown). The PLL locks on to the sampling frequency of the bit-stream 618 and produces the original frequency rate 114 of the original bit-stream. The process then ends in step 620.

While various embodiments of the invention have been described, it will be apparent to those of ordinary skill in the art that many more embodiments and implementations are possible that are within the scope of this invention.

CLAIMS

What is claimed is:

1. A method for converting a first synchronous compressed bit-stream frame of data at a first sampling rate to second synchronous compressed bit-stream frame of data at a second sampling rate, the method comprising:

determining a format for the first compressed bit-stream frame where the first compressed bit-stream frame has a frame length and the first compressed bit-stream frame includes a data-burst section having preamble section and a payload section having a payload length and a stuffing section having a stuffing length; and

zero stuffing the stuffing section responsive to determining the format.

2. The method of claim 1, wherein determining a format includes determining the payload length.

3. The method of claim 2, wherein zero stuffing includes zero stuffing the stuffing section responsive to determining the payload length.

4. The method of claim 3, wherein the stuffing length is equal to the frame length minus the payload length.

5. The method of claim 1, wherein determining a format includes determining the format from the preamble section.

6. The method of claim 5, further including

receiving the first synchronous compressed bit-stream frame of data,

determining a synchronization word and payload length from the preamble section,

comparing the first synchronous compressed bit-stream frame of data to the synchronization word,

starting a counter in response to the comparison, and

determining first sampling rate.

7. The method of claim 6, wherein zero stuffing includes zero stuffing the stuffing section responsive to determining the payload length.

8. The method of claim 7, wherein the stuffing length is equal to the frame length minus the payload length.

9. A method for converting a first synchronous compressed bit-stream frame of data at a first sampling rate having zero stuffing to second synchronous compressed bit-stream frame of data at a second sampling rate, the method comprising:

determining a format for the first compressed bit-stream frame where the first compressed bit-stream frame has a frame length and the first compressed bit-stream

frame includes a data-burst section having preamble section and a payload section having a payload length and a stuffing section having a stuffing length; and

removing zero stuffing from the stuffing section responsive to determining the format.

10. The method of claim 9, wherein determining a format includes determining the payload length.

11. The method of claim 10, wherein removing includes removing the zero stuffing from the stuffing section responsive to determining the payload length.

12. The method of claim 11, wherein the stuffing length is equal to the frame length minus the payload length.

13. A bit-stream converter for converting a first synchronous compressed bit-stream frame of data at a first sampling rate to second synchronous compressed bit-stream frame of data at a second sampling rate, the bit-stream converter comprising:

a payload length detector; and

a zero stuffing unit in signal communication with the payload length detector, the zero stuffing unit capable of zero stuffing the stuffing section responsive to the payload length detector detecting the payload length.

14. The bit-stream converter of claim 13 further including

a synchronization unit;

a frequency detector in signal communication with the synchronization unit and payload length detector; and

a counter in signal communication with the synchronization unit and zero stuffing unit.

15. An inverse bit-stream converter for converting a first synchronous compressed bit-stream frame of data at a first sampling rate having zero stuffing to second synchronous compressed bit-stream frame of data at a second sampling rate, the bit-stream converter comprising:

a synchronization unit;

a format detector in signal communication with the synchronization unit, the format detector capable of determining a format for the first synchronous compressed bit-stream frame of data; and

a zero stuffing removal unit in signal communication with format detector.

16. A bit-stream converter for converting a first synchronous compressed bit-stream frame of data at a first sampling rate to second synchronous compressed bit-stream frame of data at a second sampling rate, the bit-stream converter comprising:

means for determining a payload length; and

means for zero stuffing a stuffing section responsive to determining the payload length.

17. The bit-stream converter of claim 16 further including

means for synchronization;

a frequency detector in signal communication with the synchronization means and the determining means; and

a counter in signal communication with the synchronization means and zero stuffing means.

18. An inverse bit-stream converter for converting a first synchronous compressed bit-stream frame of data at a first sampling rate having zero stuffing to second synchronous compressed bit-stream frame of data at a second sampling rate, the bit-stream converter comprising:

means for synchronization;

means for determining a format for the first synchronous compressed bit-stream frame of data; and

means for removing the zero stuffing.

19. A signal-bearing medium having software for converting a first synchronous compressed bit-stream frame of data at a first sampling rate to second synchronous compressed bit-stream frame of data at a second sampling rate, the signal-bearing medium comprising:

logic configured for determining a format for the first compressed bit-stream frame where the first compressed bit-stream frame has a frame length and the first compressed bit-stream frame includes a data-burst section having preamble section and a payload section having a payload length and a stuffing section having a stuffing length; and

logic configured for zero stuffing the stuffing section responsive to determining the format.

20. The signal-bearing medium of claim 19, wherein determining logic includes logic configured for determining the payload length.

21. The signal-bearing medium of claim 20, wherein zero stuffing logic includes logic configured for zero stuffing the stuffing section responsive to the logic configured for determining the payload length.

22. The signal-bearing medium of claim 21, wherein the stuffing length is equal to the frame length minus the payload length.

23. The signal-bearing medium of claim 19, wherein determining logic includes logic configured for determining the format from the preamble section.

24. The signal-bearing medium of claim 23, further including

logic configured for receiving the first synchronous compressed bit-stream frame of data,

logic configured for determining a synchronization word and payload length from the preamble section,

logic configured for comparing the first synchronous compressed bit-stream frame of data to the synchronization word,

logic configured for starting a counter in response to the comparison, and

logic configured for determining first sampling rate.

25. The signal-bearing medium of claim 24, wherein zero stuffing logic includes logic configured for zero stuffing the stuffing section responsive to determining the payload length.

26. The signal-bearing medium of claim 25, wherein the stuffing length is equal to the frame length minus the payload length.

27. A signal-bearing medium having software for for converting a first synchronous compressed bit-stream frame of data at a first sampling rate having zero stuffing to second synchronous compressed bit-stream frame of data at a second sampling rate, the signal-bearing medium comprising:

logic configured for determining a format for the first compressed bit-stream frame where the first compressed bit-stream frame has a frame length and the first compressed bit-stream frame includes a data-burst section having preamble section

and a payload section having a payload length and a stuffing section having a stuffing length; and

logic configured for removing zero stuffing from the stuffing section responsive to determining the format.

28. The signal-bearing medium of claim 27, wherein the determining logic includes logic configured for determining the payload length.

29. The signal-bearing medium of claim 28, wherein removing logic includes logic configured for removing the zero stuffing from the stuffing section responsive to determining the payload length.

30. The signal-bearing medium of claim 29, wherein the stuffing length is equal to the frame length minus the payload length.

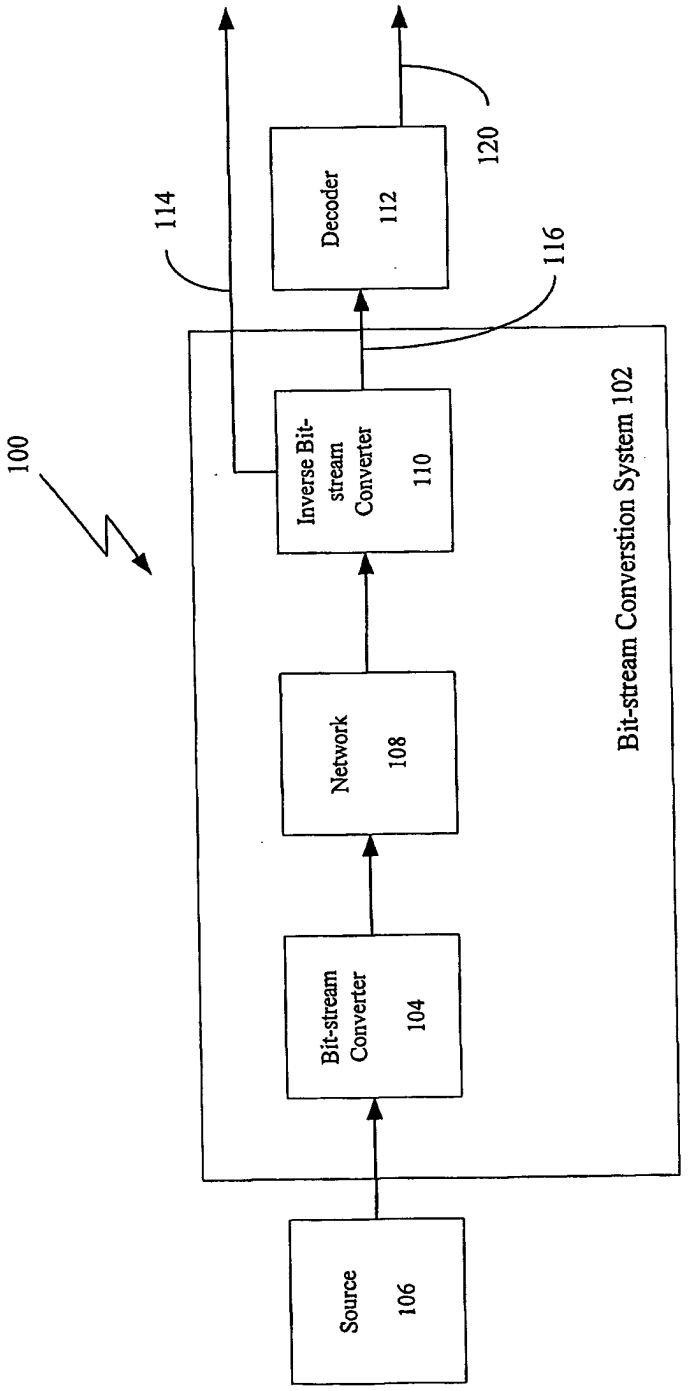
100
114
116
120
Source 106
Bit-stream Converter 104
Network 108
Inverse Bit-stream Converter 110
Decoder 112
Bit-stream Converstion System 102

FIG. 1

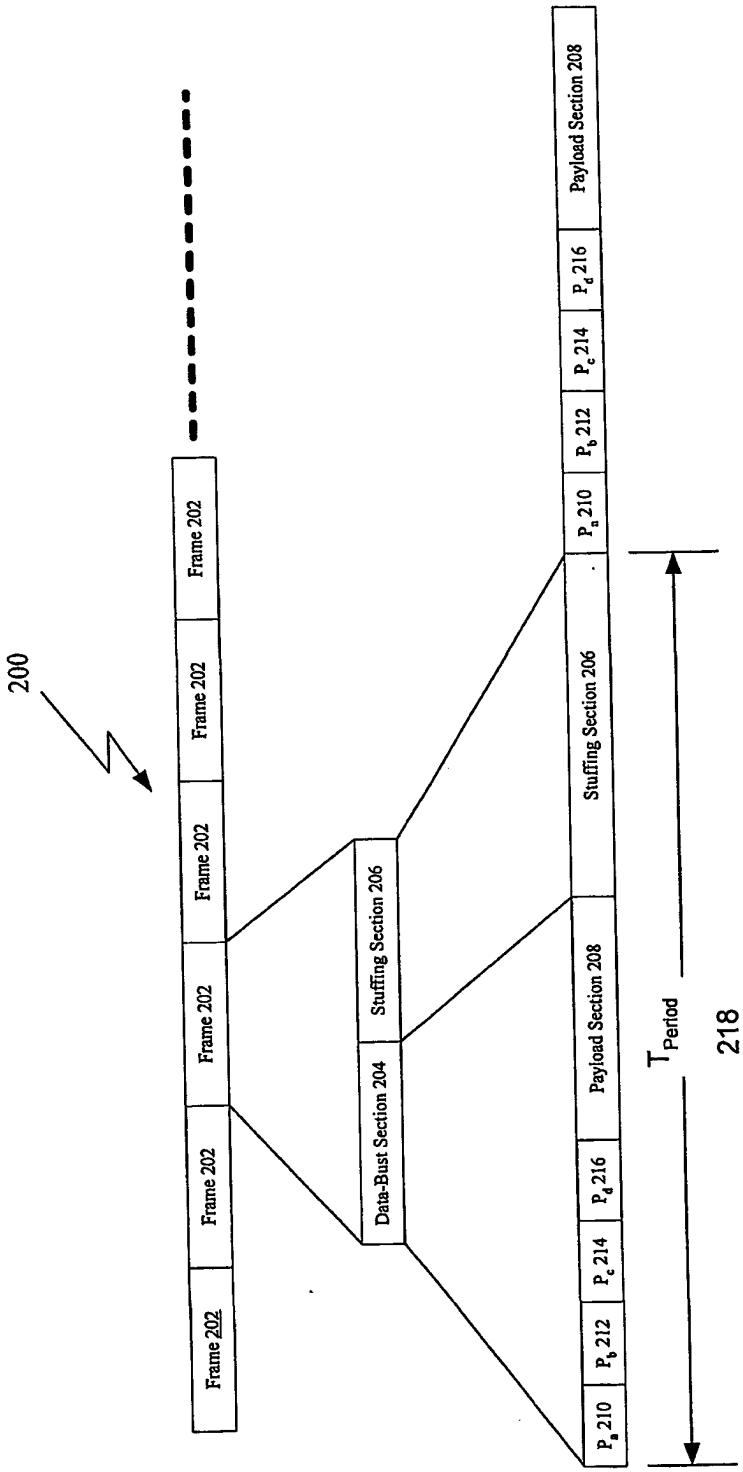
200
Frame 202
Frame 202
Frame 202
Frame 202
Frame 202
Frame 202
Data-Bust Section 204
Stuffing Section 206
Payload Section 208
P_a 210
P_b 212
P_c 214
P_d 216
Stuffing Section 206
Payload Section 208
P_a 210
P_b 212
P_c 214
P_d 216
T_{Period}
218

FIG. 2

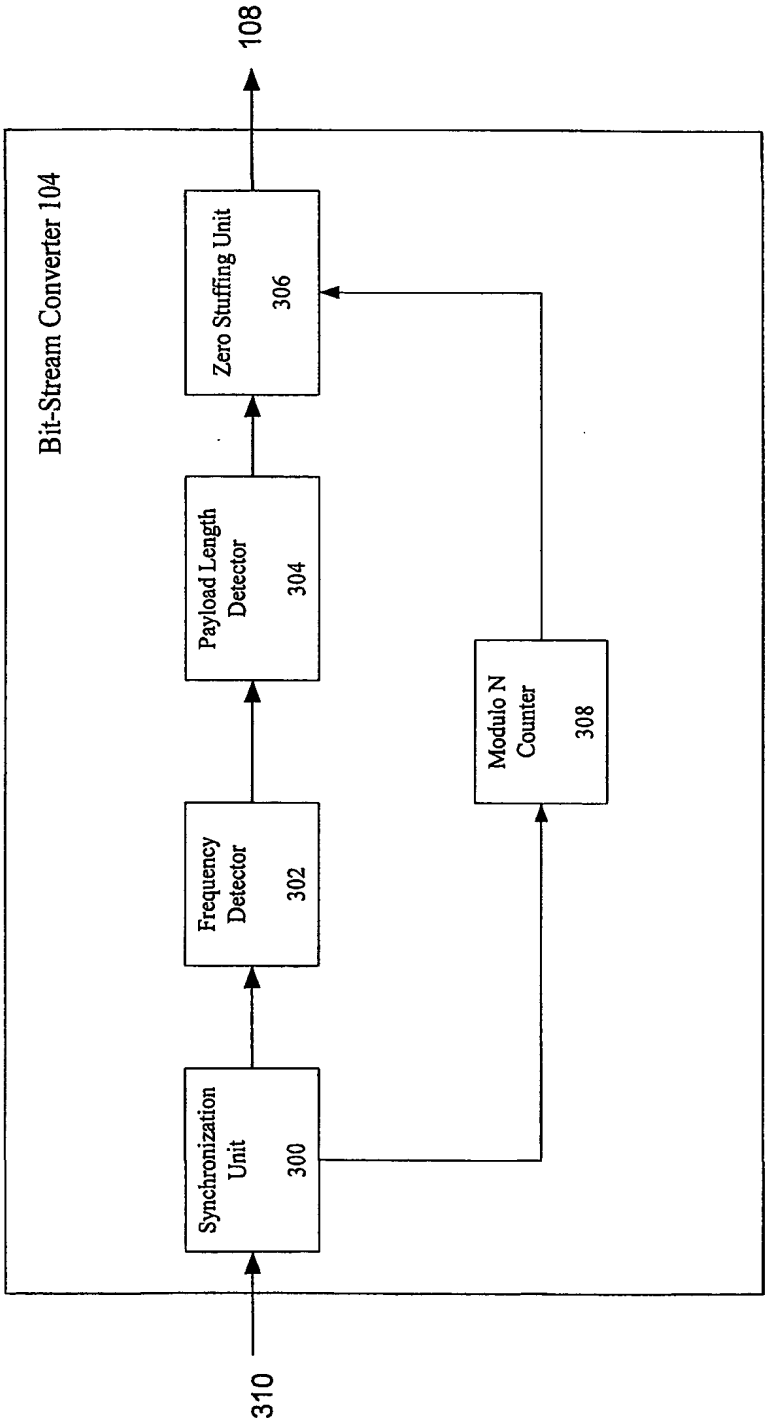
108
Bit-Stream Converter 104
Zero Stuffing Unit
306
Payload Length Detector
304
Frequency Detector
302
Synchronization Unit
300
Modulo N Counter
308
310

FIG. 3

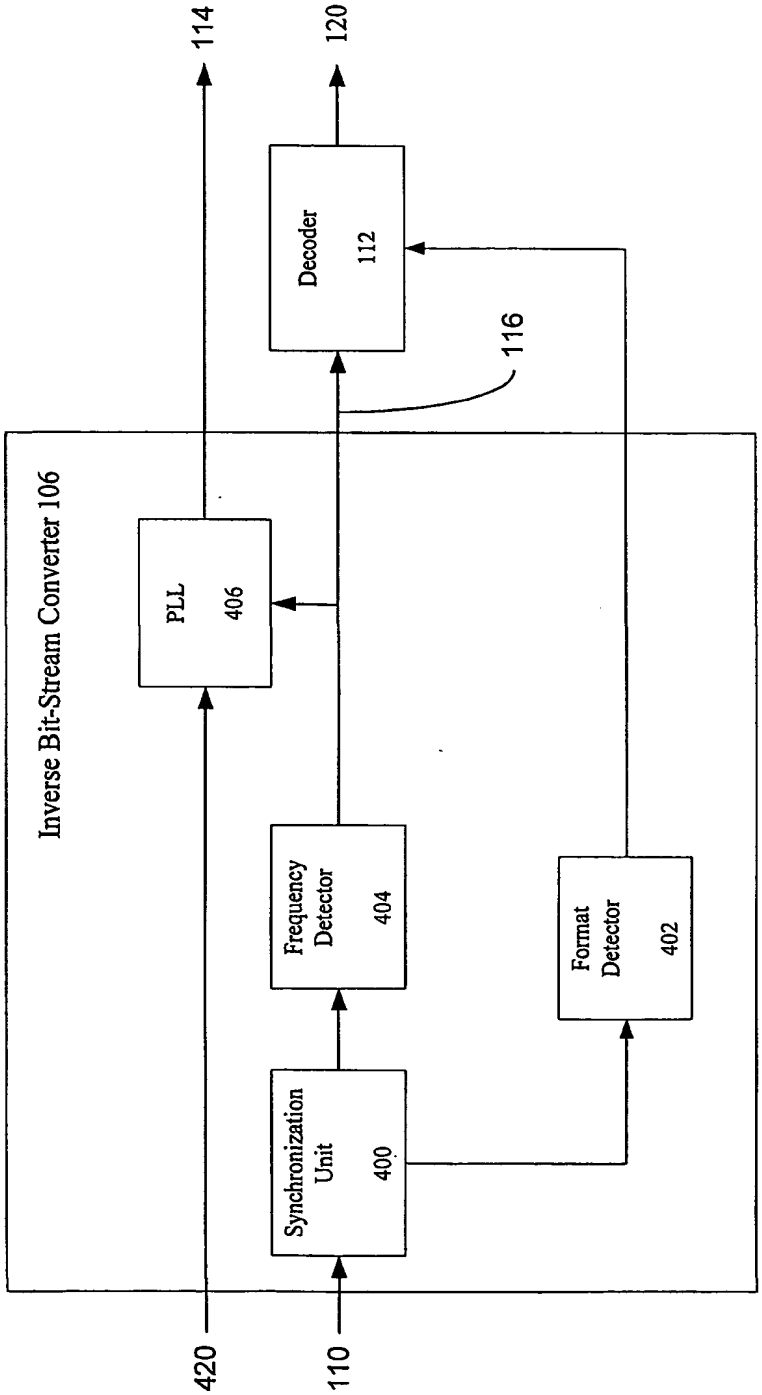
114
120
Decoder
112
116
Inverse Bit-Stream Converter 106
PLL
406
Frequency
Detector
404
Format
Detector
402
Synchronization
Unit
400
420
110

FIG. 4

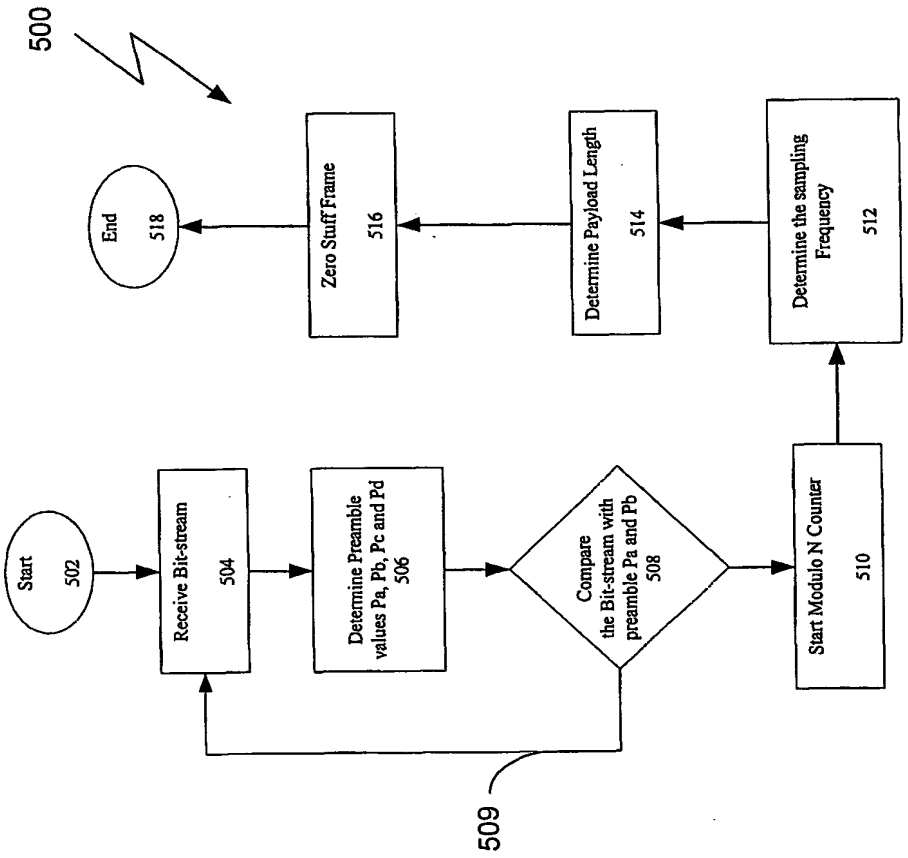
500
Start
502
Receive Bit-stream
504
Determine Preamble values Pa, Pb, Pc and Pd
506
Compare the Bit-stream with preamble Pa and Pb
508
509
Start Modulo N Counter
510
Determine the sampling Frequency
512
Determine Payload Length
514
Zero Stuff Frame
516
End
518

FIG. 5

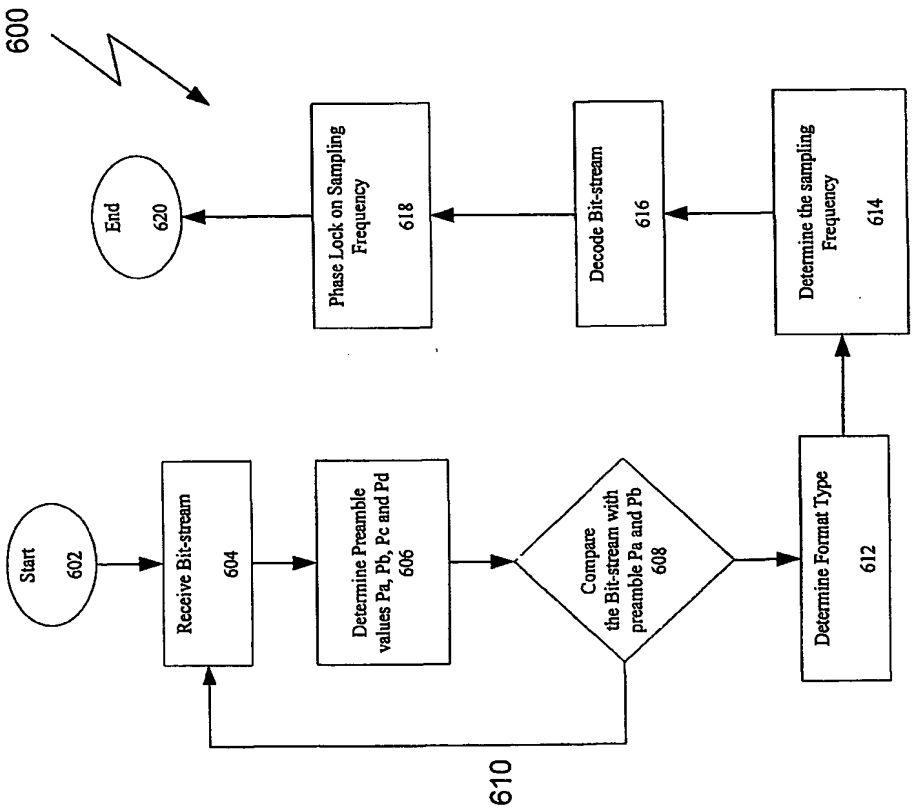
600
Start
602
Receive Bit-stream
604
Determine Preamble values Pa, Pb, Pc and Pd
606
Compare the Bit-stream with preamble Pa and Pb
608
610
Determine Format Type
612
Determine the sampling Frequency
614
Decode Bit-stream
616
Phase Lock on Sampling Frequency
618
End
620

FIG. 6